PRICE $2.00

THE 80 NOTEBOOK

JULY 1980

ISSUE #4

NOTE: The term "TRS-80" is a registered trademark of Radio Shack, a Division of Tandy Corporation. THE 80 NOTEBOOK is not affiliated with Radio Shack or Tandy Corporation in any way.

NLOS/1 AND NLOS/2

A Natural Language Operating System for the TRS-80

What is natural language? For our purposes, natural language is common, ordinary English - the expression of facts using simple sentences. NLOS/1 is a system which allows the computer to "understand" the information conveyed to it through simple sentences and to answer questions concerning the information conveyed. This ability makes the system an excellent tool for the creation, management and inquiry to a conversational data base of facts and figures. It can also be an educational tool - as a study in artificial intelligence through an examination of the internal workings of the program, or by its reaction to math reading problems of various complexity. It can be an excellent tool in teaching English grammar, sentence structure and logical deductive reasoning to students, young and old alike. In any case, holding a conversation with your computer can be a lot of fun.

Let's see how the system accomplishes this and what its limitations are:

First, we have to get down to the basics of English. The system recognizes phrases grouped together in a sentence. A phrase is a group of one or more words that, together, convey a concept or identify an entity. These phrases can convey a subject, a verb, a preposition, a conjunction, a modifier or a question invoker. These represent the grammatical types that NLOS/1 can handle. You may ask - why not recognize individual words rather than phrases so that the conventional grammatical types - nouns, adjectives, verbs, adverbs, prepositions, conjunctions, interjections, and pronouns may be used? Well, recognizing nothing smaller than a phrase alleviates the problem of context usage. For example - "the President of the United States" contains adjectives, nouns and a preposition, yet the phrase identifies a single subject. Each grammatical type further conveys a type of information. Subjects identify a person (a who - "Tom", etc.), a place (a where - "New York City", etc.), or a thing (a what - "a banana", etc.).

Verbs convey the type of object clause they affect - in "Tom said nothing" and "Tom went home", etc., "said" acts on "what" and "went" acts on "where".

Every sentence must have a verb. Prepositions introduce subjects which show what ("as a clown"), when ("on my birthday"), where ("in town"), why ("to go home") and how ("by going home").

"Why" or "how" prepositional phrases are usually a combination of prepositions and verbs (such as why - "to go to"), while most other prepositional phrases consist of a single prepositional word.

Conjunctions show no type of information but are used to combine two adjacent subject phrases into a single subject clause. Because of this, conjunctions are restricted to "and" type conjunctive phrases. Also, only one conjunction may be used in each sentence analyzed by the system. The reason for this lies in the way NLOS/1 associates one subject to another in its deductive reasoning processes. This will be discussed further when I explain the problem solving algorithm.

Modifiers are adjectives and adverbs. As adjectives, modifiers appearing immediatly before a subject relate to that subject. These show what ("red", "lazy", etc.), association ("a", "the", "their", etc. - associative modifiers are generally for aesthetic use only and are ignored by the system in its deductive reasoning), multiplication ("times as many", etc.) and numeric ("100", etc.). Multiplicative and numeric modifiers are very important in the deductive reasoning processes but numbers must appear before multiplicative modifiers in subject clauses (as in "10 times as many boxes", etc.). As adverbs, modifiers may appear before or after a verb phrase and show how ("quickly", etc.).

With all of the above taken into consideration, you can see how the system can break down a sentence into who, what, when, where, why and how catagories of information.

Simple sentences used with the system should have only one piece of information in each of these catagories.

Question invokers may be used in a sentence to request the missing catagory of information represented by the question invoker. For example - if you tell the system "Tom ran home to have dinner", you could later ask "who ran home to have dinner?" and the system would reply "Tom"; or "Tom ran where" would reply "home"; or "why did Tom run home" would reply "to have dinner" assuming that "ran" and "did run" have the same root verb - this will be discussed later.

Question invokers, therefore, can request who, what, when, where, why or how. A special class of question invoker "how many" exists which is used to invoke the deductive reasoning process.

Also, a sentence beginning with a verb phrase invokes a yes/no type of response (given "Tom shined his shoes" you could ask "did Tom shine his shoes?"). Notice that the verb phrase "did shine" has been broken up around the subject to do this. The system would respond "yes" if the information was true or "no" if the information was false or unknown.

Punctuation is not allowed within a defined phrase and is ignored in sentences. Also, numeric modifiers are recognized without formal definition and are not allowed within defined phrases. This would disallow contractions or non-integer numeric modifiers in sentences.

By means of a utility routine in the system, phrases are formally defined to and maintained by the system in a dictionary which stores the phrase, its grammatical usage and the type of information it conveys.

A phrase already in the dictionary may be redefined. To have a phrase deleted or ignored, you simply redefine it as an associative modifier since this type of phrase is for aesthetic purposes only and is ignored in sentence analysis. If a phrase is redefined, the redefinition only applies to sentences analyzed after the redefinition.

Different phrases may contain the same word or words but in different sequences or in combination with other words.

For verb phrases, the system also requires a root phrase which normalizes the time references implied by verbs. For example - "went" and "did go" are both "past go". Whenever a verb phrase is encountered in sentence analysis, it is replaced by its root phrase. This allows such analysis as "Tom went home" and "did Tom go home" with the system replying "yes".

An additional utility lists the vocabulary of phrases along with their characteristics to allow the user to review his dictionary.

As sentences are received and analyzed by the system, the information extracted is stored in a special section of the dictionary structured like an encyclopedia. This section is referenced by use of question invokers or yes/no question invokers as discussed earlier.

A special note - a simple form of compound sentences can be handled by the system during sentence analysis.

A noun followed by a verb may introduce another sentence (as in "Tom said Dick went home"). In this example the system treats the one sentence as two statements: "Tom said what (Dick went home)" and "Dick went home".

The system also contains a utility routine for scrolling thru the encyclopedia section of the dictionary to allow the user to view the information the system has thus far received, analyzed and cataloged. This display shows the information in the sentences and the "information type" catagories they were assigned. The sentences are shown in the sequence they were received to show the flow of changes in facts.

Another function of sentence analysis, which is critical to the deductive reasoning process, is the association of one subject in a sentence relative to the other subjects in the sentence.

For example - in "Tom has 3 balls in a box", two relationships are extracted: "Tom" with "3 balls" and "3 balls" with "box". If you were to ask "how many balls has Tom", the system would reply "3". Notice that the "how many" questioning sentence can relate only one subject to another and can not be conditioned by any how, why or when type clauses.

If a conjunction is used as in "Tom and Dick have a car", the relationships extracted are: "Tom" with "car" and "Dick" with "car"; but, in "Tom and Dick have a car and a truck", if two conjunctions were allowed, the relationships would be "Tom" with "car", "Dick" with "car", "Tom" with "truck" and "Dick" with "truck".

The system can only handle up to 3 subjects in a relationship at one time; so, only one conjunction is allowed per sentence.

These relationships are also stored in a special section of the dictionary structured to handle one to one associations. Along with each of the two subjects in a relationship, the dictionary also stores the numeric modifiers, if any, and the multiplicative modifiers, if any, associated with those subjects. Both subjects in a relationship may not contain multiplicative modifiers and association dictionary entries are not built if the sentence has how, why or when type clauses because the system can not handle conditional relationships in its deductive reasoning processes. For example - in "Tom ran quickly to the store" would not relate "Tom" with "store" because of the how "quickly". Also, "3 times as many balls are used for 2 times the boxes" because of the abstract nature of the relationship between balls and boxes.

And now for the deductive reasoning process - let's consider the following sample problem: given the following dictionary of phrases - a when preposition "on", an associative modifier "their", a what subject "vacation", a who subject "Tom", a who subject "Dick", a where verb "visited" with root "past visit", an associative modifier "a", a where subject "farm", a when preposition "while", a where subject "there", a who subject "he", a what verb "noticed" with root "past notice", a what subject "pen", a what verb "containing" with root "present contain", a what subject "chickens", a what subject "pigs", a what verb "said" with root "past say", a who subject "they", a what verb "were" with root "past be", a multiplicative modifier "times as many", a what preposition "as", a what verb "counted" with root "past count", a what subject "legs", a where preposition "in", an associative modifier "the", a what verb "have" with root "present have", a conjunction "and", and a how many question invoker "how many"; the following sentences can be analyzed by the system - "chickens have 2 legs", "pigs have 4 legs", "on their vacation Tom and Dick visited a farm", "while there they noticed a pen containing chickens and pigs", "Tom said there were 3 times as many chickens as pigs", "Dick said he counted 100 legs in the pen" and "how many chickens were in the pen". The system would reply to the last questioning sentence - "30".

How could the system have deduced this? Let's look at the approach the system takes: the system would first make all the subject to subject relationships possible - "chickens" with "2 legs", "pigs" with "4 legs", "pen" with "chickens", "pen" with "pigs", "Tom" with "there", "there" with "3 times as many chickens", "3 times as many chickens" with "pigs", "Dick" with "he", "he" with "100 legs" and "100 legs" with "pen". Since "pen" is the area of concern in which to search for "how many", the relationships involving "pen" are selected - "chickens", "pigs", "100 legs".

If a non-multiplicative numeric modifier had been associated with chickens, that number would be the answer (the system accepts facts given it as truth; but, if the system had been asked "how many chickens were in 3 pens", the answer found would be multiplied by 3). The system must now look for a relationship with a non-multiplicative numeric modifier to give "pen" a numerical base of equivalence. This relationship would be "100 legs". Since "100 legs" is to be treated as "pen", the remaining "pen" relationships are matched against all of the other relationships in the dictionary in order to pick out these relationships with "legs". This would result in the selection of the following relationships -

"chickens" with "2 legs" and "pigs" with "4 legs". The system must also check for any relationships between the resultant subjects. This search shows the relationship of "3 times as many chickens" with "pigs".

With all of this selected information, an equation can be formulated - "100 legs" equals "3 times as many chickens" plus "pigs". Substituting the number of legs represented by chickens and pigs, a common unit of measure can be applied to the equation. This yields the equation - "100 legs" equals "3 times as many" times "2 legs" plus "4 legs" or 100 equals 6X plus 4X. Then X equals 10 and since there are "3 times as many chickens", there must be 3 times 10 or 30 chickens. At this point, the system would simply print "30" and ask for another sentence.

The above algorithm can be applied to a wide variety of math reading style problems. It can be an educational experience just exploring the many variations possible.

The dictionary you construct and its component phrases, sentences and subject relationships can be stored as a file on cassette tape and then reloaded at the beginning of a session. This allows an accumulated data base of information to be maintained for inquiry purposes.

Operationally, NLOS/1, a 16K Level 2 BASIC program, is divided into sections which flow one to another. To begin with, the program asks if you want it to read a dictionary tape created in a previous run. This allows you to build and retain dictionary tapes on various problems and subjects and to re-use this information as needed by having the system re-input the information from tape in a later run. Enter "yes" if you wish a previously created dictionary tape to be read or enter "no" to go on.

After you have entered "no" or the dictionary tape has been read, the system will ask if you are defining a phrase. This section of the program allows the user to add phrase definitions to the system dictionary in memory or to redefine a phrase previously known to the system.

After the phrase definition is entered, this section of the program will repeat until a "no" response is entered to the "DEFINING PHRASES" question. If you enter "yes" to the "DEFINING PHRASES" prompt, you will be expected to answer a series of questions which will supply the information needed to define a phrase to the system.

The first thing the system must know is the phrase text itself. Next, it must be told if the phrase is a 1) subject, 2) verb, 3) preposition, 4) conjunction, 5) modifier, or 6) question invoker by entering the appropriate code.

Once this grammatical usage information is known about a phrase, the system will ask for the type of information that the phrase will convey when used in a sentence. You must enter the proper code from the following list: 1) who, 2) what, 3) when, 4) where, 5) how, 6) how many, 7) association, 8) multiplicative, 9) why.

As discussed previously, the allowable type of information conveyed by a phrase is determined by the grammatical usage. If an improper type of information code is entered, the type of information question will be repeated.

If the phrase being defined is a verb you must next enter the verb phrase which must have been defined which represents the root verb phrase. As discussed previously, this is required to commonize the various phrases which have the same functional usage.

If you wish to delete a phrase from the system's dictionary, simply redefine it with a modifier grammatical usage and an association information type. This will cause the phrase to be ignored during sentence analysis since this class of phrases normally includes words such as "the" or "an" which are functionally unneeded in a sentence.

Once the user has defined all of the phrases needed, you will be given the opportunity to list all the phrases in the dictionary as a review. This will happen if you enter "yes" to "WANT VOCABULARY LIST". If you enter "no" the program will go on.

After this you will be given the opportunity to review the sentences previously inputted if a dictionary tape was loaded by entering "yes" to the "LIST ENCYCLOPEDIA" prompt. These sentences will be displayed in a form broken down into its information type components.

After all of this maintenance and review activity has taken place, the user may begin entering sentences conveying various information by combining phrases previously defined to the system using simple English grammar and enter questions made up of previously defined phrases which ask about information conveyed in previously entered sentences or calculate unknown numerical statistics from quantitative information and relationships conveyed in previously inputted sentences.

The system recognizes a sentence as a question if it begins with a question invoker phrase or if it begins with a verb phrase, in which case the question is taken to be a yes/no confirmation of an informational sentence previously inputted.

Each sentence or question must be entered one at a time in response to the "ENTER SENTENCE OR HIT ENTER". If you hit enter only in response to this prompt, the program will go on to the last section of the system.

During sentence analysis, several error messages may appear. "UNRECOGNIZABLE PHRASE" means that the present sentence being analyzed contains a phrase not in the current dictionary. Any sentences inputted prior to this message are saved in the dictionary. The user can hit the BREAK key and type in "GOTO 455" (ENTER) to put the program back into the "DEFINING PHRASES" section so that the missing phrase can be defined. After doing this and going through the program until you get back to the "ENTER SENTENCE" prompt, you may re-enter the sentence which caused the original error message.

The "INVALID GRAMMAR" message occurs if the sentence being analyzed is too complex for the system. The user should re-enter the same information but using one or more simpler sentences which collectively express the same thought.

"UNKNOWN" and "TOO COMPLEX" messages occur when a "how many" question is too difficult or insufficient data exists in the dictionary. Additional informational sentences may be needed to be inputted before the question is retried or the question might need to be re-expressed in a different form that the system may understand better.

Finally, the last section of the program writes the dictionary of phrase definitions and sentences to cassette tape if the user responds "yes" to the "SAVE DICTIONARY ON TAPE" prompt. If you respond "no" or after the tape is written, the program ends.

It should be noted that the format of the dictionary uses all string storage. Because of this, the system does not allow punctuation in sentences and numeric modifiers involved in subject associations may only contain non-negative integer numbers. Also, sentence analysis may require several minutes, especially involving "how many" question computation.

In contrast, let us look at the improvements found in NLOS/2 not found in NLOS/1. First, NLOS/2 is a larger, more involved program requiring 32K. The dictionary format has been changed to allow faster execution and more conservation in storage. One drawback to this is that NLOS/2 can not read an NLOS/1 dictionary tape.

Numeric modifiers may contain decimal points and negative values to allow for greater computational flexibility.

Rather than sequentially running through each functional section of the program as we did in NLOS/1, NLOS/2 allows the user to select the section desired by use of a series of commands. Rather than using sectional (yes/no availability) prompts, NLOS/2 uses a general command prompt of a right arrow. The response to this prompt should be "DEFINITION" or "DEF" to initiate a phrase definition, "VOCABULARY" to list the phrase definition portion of the dictionary, "SENTENCE" to list the sentences currently stored in the dictionary, "END" to write the dictionary to tape and end the program, and "PROCEDURE" to input or update the subroutine associated with an action verb phrase.

If the text entered at the arrow prompt is other than one of the above commands, it is assumed to be a sentence or question to be analyzed.

Concerning action verb subroutines maintained by the procedure command, if a question starts with a verb phrase with a subroutine, the sentence is passed to the subroutine and the subroutine is executed as if it were an order or command with the remainder of the sentence treated as a parameter to be interrogated by the subroutine, rather than the sentence being treated as a yes/no question as is implied in a verb phrase with no subroutine associated with it.

As you can see, NLOS/2 has some large advantages as compared to NLOS/1. Some of the other smaller changes include the ability to configure the maximum number of phrase definitions, sentence storage and subject association entries allowed in the dictionary as opposed to NLOS/1 which has a fixed limit on the size of the dictionary; and an instructional subroutine which teaches the user how to set up the coding structure needed for an action verb phrase subroutine to function.

If you would like to run NLOS/2 in only 16K, you would have to give up the action verb subroutine capability. This can be done by using the following changes:

```
DELETE 1600-1945
DELETE 7500-7585
DELETE 9800-9974

1600 RETURN
7500 RETURN
9800 GOTO 1140
```

This would give you a super NLOS/1 with the storage and execution advantages of NLOS/2.

And now, here is the programming listing for NLOS/2:

```
1 CLS:PRINTCHR$(23):PRINTTAB(14)"NLOS":PRINT:PRINTTAB(15)"A":PRINTTAB(8)"NATURAL
 LANGUAGE":PRINTTAB(8)"OPERATING SYSTEM":PRINT
2 PRINTTAB(9)"COPYRIGHT 1979":PRINTTAB(10)"CYBERMATE CO.":PRINTTAB(8)"R.D.#3 BOX
 192A":PRINTTAB(6)"NAZARETH, PA  18064":PRINTTAB(7)"PHONE 215-759-6873"
3 PRINT:PRINT"  CASSETTE $4.95, SOURCE $1.95":PRINT:INPUT"NEED INSTRUCTIONS(Y/N)"
,V$:IFV$="Y"THENGOSUB1600
4 CLS:PRINT"I NEED TO KNOW SOME STATISTICS":DEFINTA,B,C,I
5 INPUT"HOW MUCH STRING SPACE MAY I USE";J:CLEARJ:I4=32:IL=4:IK=10
6 INPUT"WHAT IS THE MAXIMUM NUMBER OF DIFFERENT PHRASES THAT YOU WILL TEACH ME";
I1:IFI1<10THENI1=10
7 INPUT"WHAT IS THE MAXIMUM NUMBER OF SENTENCES USING THOSE PHRASES THAT YOU WIL
L INPUT TO INFORM ME ON TOPICS";I2:IFI2<10THENI2=10
8 INPUT"WHAT IS THE MAXIMUM NUMBER OF SUBJECT ASSOCIATIONS I MAY LEARN ABOUT";I3
:IFI3<10THENI3=10
9 DIMA0$(I1),A1(I1,4),A2$(I2,8),A3$(I3,2),A4(I3,2),A5!(I4),A6$(I4),H(IK,IL),A8!(
I3,2),A9(I2)
10 I9=(MEM-256)/5:DIMA7(I9):PRINT"THANK YOU!"
11 FORI6=0TOI1:A0$(I6)=" ":FORI7=0TO4:A1(I6,I7)=0:NEXTI7:NEXTI6:C1=0
12 FORI6=0TOI2:FORI7=0TO8:A2$(I6,I7)=" ":NEXTI7:A9(I6)=0:NEXTI6:C2=0
13 FORI6=0TOI3:FORI7=0TO2:A3$(I6,I7)=" ":A4(I6,I7)=0:A8!(I6,I7)=0:NEXTI7:NEXTI6:
C3=0
14 FORI6=0TOI4:A5!(I6)=0:A6$(I6)=" ":NEXTI6
15 FORI6=0TOI9:A7(I6)=-1:NEXTI6:C9=0
16 FORI6=0TOIK:FORI7=0TOIL:H(I6,I7)=0:NEXTI7:NEXTI6:V2$=" "
18 PRINT"YOU HAVE A TOTAL OF ";I9;" INSTRUCTION SPACES FOR PROCEDURES!"
25 INPUT"SHOULD I LOAD A DATA BASE TAPE(Y/N)";V$:IFV$<>"Y"THEN1140
30 INPUT#-1,C1,C2,C3,BH
40 IFC1<1THEN50
45 FORI6=1TOC1:INPUT#-1,A0$(I6),A1(I6,1),A1(I6,2),A1(I6,3),A1(I6,4):A0$(I6)=" "+
A0$(I6):NEXTI6
50 IFC2<1THEN60
55 FORI6=1TOC2:INPUT#-1,A2$(I6,1),A2$(I6,2),A2$(I6,3),A2$(I6,4),A2$(I6,5),A2$(I6
,6),A2$(I6,7),A2$(I6,8),A9(I6):NEXTI6
56 FORI6=1TOC2:FORI7=1TO8:A2$(I6,I7)=" "+A2$(I6,I7):NEXTI7:NEXTI6
60 IFC3<1THEN70
65 FORI6=1TOC3:INPUT#-1,A3$(I6,1),A3$(I6,2),A4(I6,1),A4(I6,2),A8!(I6,1),A8!(I6,2
):NEXTI6
66 FORI6=1TOC3:FORI7=1TO2:A3$(I6,I7)="  "+A3$(I6,I7):NEXTI7:NEXTI6
70 IFBH<1THEN1140
72 C9=0
75  FORI6=1TOBH
80 INPUT#-1,B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,BA,BB,BC,BD,BE,BF,BG
85 IFB0<>-1THENC9=C9+1:A7(C9)=B0
90 IFB1<>-1THENC9=C9+1:A7(C9)=B1
95 IFB2<>-1THENC9=C9+1:A7(C9)=B2
100 IFB3<>-1THENC9=C9+1:A7(C9)=B3
105 IFB4<>-1THENC9=C9+1:A7(C9)=B4
110 IFB5<>-1THENC9=C9+1:A7(C9)=B5
115 IFB6<>-1THENC9=C9+1:A7(C9)=B6
120 IFB7<>-1THENC9=C9+1:A7(C9)=B7
125 IFB8<>-1THENC9=C9+1:A7(C9)=B8
130 IFB9<>-1THENC9=C9+1:A7(C9)=B9
135 IFBA<>-1THENC9=C9+1:A7(C9)=BA
140 IFBB<>-1THENC9=C9+1:A7(C9)=BB
145 IFBC<>-1THENC9=C9+1:A7(C9)=BC
150 IFBD<>-1THENC9=C9+1:A7(C9)=BD
155 IFBE<>-1THENC9=C9+1:A7(C9)=BE
160 IFBF<>-1THENC9=C9+1:A7(C9)=BF
165 IFBG<>-1THENC9=C9+1:A7(C9)=BG
170 NEXTBH:GOTO1140
200 IFN=2THENB1=A1(I6,4)
205 RETURN
460 MB$=" ":INPUT"PHRASE";MB$:J=LEN(MB$)
469 MA$=" ":I7=0:B1=0
470 N=0:PRINT"1=SUBJECT, 2=VERB, 3=PREPOSITION, 4=CONJUNCTION, 5=MODIFIER, 6=QUESTION
 INVOKER":INPUTN:IFN<1ORN>6THEN470
```

```
480 K=7:ONNGOTO490,500,510,560,530,550
490 INPUT"1=WHO,2=WHAT,4=WHERE";K:IFK=1ORK=2ORK=4THEN560ELSE490
    INPUT"2=WHAT,4=WHERE";K:IFK=2ORK=4THEN560ELSE500
    INPUT"2=WHAT,3=WHEN,4=WHERE,5=HOW,9=WHY";K:IFK=2ORK=3ORK=4ORK=5ORK=9THEN560E
LSE510
530 INPUT"2=WHAT,5=HOW,7=ASSOCIATION,8=MULTIPLICATIVE";K:IFK=2ORK=5ORK=7ORK=8THE
N560ELSE530
550 INPUT"1=WHO,2=WHAT,3=WHEN,4=WHERE,5=HOW,6=HOW MANY,9=WHY";K:IFK=1ORK=2ORK=3O
RK=4ORK=5ORK=6ORK=9THEN560ELSE550
560 IFN<>2THEN600
575 IFC1<1THEN612
580 INPUT"ROOT VERB PHRASE";MA$
582 FORI6=1TOC1:IFA0$(I6)=MA$THENI7=I6
583 NEXTI6
600 IFC1<1THEN612
610 FORI6=1TOC1:IFA0$(I6)=MB$THENGOSUB200:GOTO620
611 NEXTI6
612 I6=C1+1:IFI6>I1THENPRINT"TOO MANY PHRASES!":RETURN
620 A0$(I6)=MB$:A1(I6,1)=N:A1(I6,2)=K:A1(I6,3)=I7:A1(I6,4)=B1:C1=I6
621 IFI7=0THENA1(I6,3)=I6
622 RETURN
775 INPUT"PHRASE";MB$:I6=1:IFMB$="ALL"ANDC1>0THEN790
776 IFC1<1THENRETURN
777 FORI6=1TOC1:IFA0$(I6)=MB$THEN790
    NEXTI6:RETURN
779 IFI6>C1THENRETURN
790 N=A1(I6,1):K=A1(I6,2):I7=A1(I6,3):I8=A1(I6,4)
800 PRINT"PHRASE-";A0$(I6)
840 IFN=1V$="SUBJECT"
850 IFN=2V$="VERB"
860 IFN=3V$="PREPOSITION"
870 IFN=4V$="CONJUNCTION"
880 IFN=5V$="MODIFIER"
900 IFN=6V$="QUESTION INVOKER"
920 PRINT"GRAMMAR-";V$:P=P+1:IFK=1V$="WHO"
940 IFK=2V$="WHAT"
950 IFK=3V$="WHEN"
960 IFK=4V$="WHERE"
970 IFK=5V$="HOW"
980 IFK=6V$="HOW MANY"
990 IFK=7V$="ASSOCIATION"
1000 IFK=8V$="MULTIPLICATIVE"
1015 IFK=9V$="WHY"
1020 PRINT"USAGE-";V$:IFN=2THENPRINT"ROOT VERB PHRASE-";A0$(I7)
1028 IFMB$="ALL"THENINPUT"HIT ENTER TO CONTINUE";V$:I6=I6+1:GOTO779
```

```
1029 RETURN
1140 V$=" ":INPUT">";V$:P=0
1141 IFV$="PROCEDURE"THENGOSUB7500:GOTO1140
1142 IFV$="DEFINITION"THENGOSUB460:GOTO1140
1143 IFV$="VOCABULARY"THENGOSUB775:GOTO1140
1144 IFV$="DEF"THENGOSUB460:GOTO1140
1145 IFV$="SENTENCE"THENGOSUB8500:GOTO1140
1146 IFV$="END"ORV$=" "ORV$=""THEN8000
1149 V$=V$+"  "
1150 J=LEN(V$):Z8=0:T=0:Z0$=" ":B$=" "
1151 P1=0:P2=0
1160 ZH=0:ZI=0:ZQ=0:N=0:K=0:L=0:ZB=0:ZE=0:XM=0:ZZ=0
1165 M1$=" ":M2$=" ":M3$=" ":M4$=" ":M5$=" ":M6$=" ":M7$=" ":E$=" ":ZA=0:B$=" ":
ZY=0:Q$=" "
1167 S1$=" ":S2$=" ":S3$=" ":S4$=" ":V1=0:V2=0:V3=0:V4=0:X1=0:X2=0:X3=0:X4=0:D$=
" "
1168 SR=0
1170 P=P+1:IFP>JTHEN6000
1180 F1=ASC(MID$(V$,P,1)):IFF1=32THEN1170
1181 IFSR=0THENSR=P
1182 IFF1=43ORF1=45ORF1=46THEN1196
1190 IFF1<48ORF1>57THEN1240
1196 SL=1
1198 SR=P
1200 P=P+1:IFP>JTHEN1230
1220 F1=ASC(MID$(V$,P,1)):IFF1=32THEN1230
1221 SL=SL+1:GOTO1200
1230 C$=MID$(V$,SR,SL)
1231 T=VAL(C$):N=5:K=10
1232 S9=0:SR=1:SL=1
1233 GOTO1340
1240 SR=-1:SL=0:IFC1<1THEN1250
1241 FORI6=1TOC1
1242 IFLEN(V$)-P+1<LEN(A0$(I6))THEN1246
1243 SQ=LEN(A0$(I6))
1245 IFA0$(I6)=MID$(V$,P,SQ)ANDSQ>SLTHENSR=I6:SL=SQ
1246 NEXTI6
1250 IFSR=-1THEN2222
1260 C$=A0$(SR):N=A1(SR,1):K=A1(SR,2):S9=SR:SR=SL
1340 IFN=1ANDZH<3ANDZB=1THEN2222
1341 IFZH=9ANDN<>4THENZH=0:GOSUB3070
1342 IFN=4ANDZB<>1THEN2222
1343 IFZH=3ORZH=9THEN4100
1344 IFN=4ANDZ8=1THEN2222
1345 IFN=4ANDZQ=7THEN2222
```

```
1346 IFZH=3THEND$=D$+" "+C$
1347 IFZB=4ANDN<>1THEN2222
1350 IFN=5ANDK<7THEND$=D$+" "+C$
1355 IFN=5B$=B$+" "+C$
1360 IFN=3THEND$=C$:ZH=3:ZI=K
1370 IFZH=3ANDN=1ZH=9
1390 IFN=6THEN2020
1400 IFN=5ANDK=8XM=1
1405 IFN=5ANDK=8ANDI<2THEN2222
1410 IFN=2ANDZA=0ANDZ0=0Z0=8
1415 IFZ0=8ANDA1(S9,4)>0THEN3000
1420 IFN=3ANDZH=2ANDQ$<>" "M6$=M6$+"/"+Q$:Q$=" ":B$=" "
1425 IFN=1THENE$=Q$+" "+C$:Q$=" "
1430 IFN=1ANDB$<>" "C$=B$+" "+C$:Q$=" ":B$=" "
1435 IFZH<3ANDN=1GOSUB4300
1440 IFN=2ANDQ$<>" "M6$=M6$+"/"+Q$:Q$=" ":B$=" "
1460 IFN=4Z8=1
1472 IFN=2ANDZ0=0ANDP1>0THENZ0$=MID$(V$,P1):Z2=P2:GOSUB2135:P=P1:GOTO1150
1480 IFN=2GOSUB3180
1490 IFN=1ANDZB=3GOSUB4600
1500 IFN=1ANDZB=2GOSUB4600
1510 IFN=1ANDZB=4GOSUB4800
1520 IFN=1ANDZB<>4GOSUB4700
1530 IFN=1I=0:XM=0:ZY=ZY+1
1540 GOTO2000
1600 CLS:PRINT"NLOS/2 USERS MUST HAVE A WORKING KNOWLEDGE OF NLOS/1!"
1695 PRINT"WHEN THE > PROMPT APPEARS, YOU MUST ENTER A COMMAND OR"
1700 PRINT"A SENTENCE OR QUESTION. THE COMMANDS ARE AS FOLLOWS-"
1705 PRINT"DEFINITION - ALLOWS YOU TO DEFINE A PHRASE TO THE SYSTEM"
1710 PRINT"VOCABULARY - ALLOWS YOU TO EXAMINE A PHRASE DEFINITION OR"
1715 PRINT"ENTER ALL TO EXAMINE ALL PHRASES. SENTENCE - ALLOWS YOU TO"
1720 PRINT"EXAMINE SENTENCES PREVIOUSLY INPUTTED. PROCEDURE - ALLOWS"
1725 PRINT"YOU TO DEFINE A BASIC PROGRAM TO BE RUN WHEN THE SYSTEM"
1730 PRINT"SEES A PARTICULAR VERB PHRASE. THIS INVOLVES ENTERING A"
1735 PRINT"SERIES OF OP CODES AND THEIR OPERANDS WHICH THE SYSTEM WILL"
1740 PRINT"TRANSLATE INTO A BASIC PROGRAM. THE SYSTEM SUPPLIES WORK"
1745 PRINT"SPACE FOR YOUR BASIC ROUTINE IN THE FORM OF 33 SINGLE"
1750 PRINT"PRECISION FIELDS NUMBERED 0 THRU 32 AND 33 STRING FIELDS"
1755 PRINT"NUMBERED 0 THRU 32. THE WORK SPACE IS INITIALIZED PRIOR"
1760 PRINT"TO RUNNING THE VERB PROCEDURE EXCEPT FOR STRING 0 WHICH"
1762 INPUT"HIT ENTER TO CONTINUE";V$:CLS
1765 PRINT"CONTAINS THE SENTENCE CONTAINING THE VERB PHRASE JUST"
1770 PRINT"INPUTTED. THE VERB PROCEDURE IS INPUTTED INTO A BUFFER"
1775 PRINT"ALLOCATED FROM CONTIGUOUS INTEGER FIELDS. YOU MUST TELL"
1780 PRINT"THE SYSTEM HOW MANY INTEGER SPACES TO ALLOW FOR A VERB."
1785 PRINT"EACH SPACE IS ADDRESSED NUMERICALLY FROM 1 TO N. FOR A NEW"
1790 PRINT"PROCEDURE, ALL SPACES ARE INITIALIZED TO 0. SPACES MAY BE"
1795 PRINT"LISTED, CHANGED IN VALUE OR HAVE THE ASC VALUES OF A"
1800 PRINT"STRING INSERTED STARTING AT A SPECIFIED LOCATION WITH"
1805 PRINT"THE STRING LENGTH INSERTED BEFORE THE STRING. THIS IS"
1810 PRINT"USED TO INPUT STRING OPERANDS FOR OP CODES THAT REQUIRE"
1815 PRINT"THEM. ANY OPERANDS REQUIRED BY AN OP CODE MUST FOLLOW"
1820 PRINT"THAT OP CODE IN THE INSTRUCTION SPACE. HERE ARE THE OP"
1825 PRINT"CODES AND THEIR OPERAND REQUIREMENTS-"
1830 INPUT"HIT ENTER TO CONTINUE";V$:CLS
1835 PRINT"OP CODE-NAME,OPERANDS (S-STRING,N-NUMERIC FIELD,R-STRING"
1840 PRINT"FIELD,T-NUMERIC STRING,L-LABEL NUMBER)"
1845 PRINT"0-NOP 1-CLS 2-LET,R,S 3-LET,N,T 4-IF,R,R 5-IF,N,N 6-POKE,N,N"
1850 PRINT"7-PEEK,N,N 8-ADD,N,N 9-SUBTRACT,N,N 10-MULTIPLY,N,N"
1855 PRINT"11-DIVIDE,N,N 12-[,N,N 13-$+$,R,R 14-INT,N,N"
1860 PRINT"15-SIN,N,N 16-COS,N,N 17-TAN,N,N 18-ATN,N,N 19-VAL,N,R"
1865 PRINT"20--STR$,R,N 21-STRING$,R,N,R 22-LPRINT,R 23-INKEY$,R"
1870 PRINT"24-JUMP TO LABEL,L 25-ABS,N,N 26-INP,N,N 27-PRINT,R"
1875 PRINT"28-PRINT@,N,R 29-INPUT,R 30-INPUT#-1,R 31-PRINT#-1,R"
1880 PRINT"32-LEFT$,R,R,N 33-RIGHT$,R,R,N 34-MID$,R,R,N,N 35-MOVE,N,N"
1885 PRINT"36-MOVE,R,R 37-EXP,N,N 38-LOG,N,N 39-JUMP IF =,L"
1890 PRINT"40-JUMP IF >,L 41-JUMP IF <,L 42-ASC,N,R 43-CHR$,R,N"
1895 PRINT"44-LEN,N,R 45-SQR,N,N 46-RETURN 47-PRINT@,N 48-SET,N,N"
1900 PRINT"49-RESET,N,N 50-POINT,N,N,N 51-OUT,N,N 52-RND(0),N"
1902 INPUT"HIT ENTER TO CONTINUE";V$:CLS
1905 PRINT"53-INSTR,R,R,N 54-END 55-GOSUB,L 999-LABEL,L"
1910 PRINT"MOST OP CODES AND THEIR OPERANDS OPERATE LIKE THEIR BASIC"
1915 PRINT"COUNTERPARTS WITH THE FIRST OPERAND BEING THE RESULT FIELD."
1920 PRINT"POINT AND INSTR PUT 1 IN THE RESULT NUMERIC FIELD WITH"
1925 PRINT"THE FIRST OPERAND OF INSTR BEING THE LARGER STRING IN THE"
1930 PRINT"SEARCH. A JUMP SHOULD FOLLOW AN IF INSTRUCTION."
1935 PRINT"YOU SHOULD NOT ATTEMPT TO PROGRAM A VERB PROCEDURE UNLESS"
1940 PRINT"YOU ARE EXPERIENCED IN LEVEL 2 BASIC PROGRAMMING!"
1945 INPUT"HIT ENTER TO CONTINUE";V$:CLS:RETURN
2000 P=P+SR-1:IFN=2P1=P+1
2001 IFN<>5ZB=N:ZE=K
2002 IFN=2P2=ZI
2010 GOTO1170
2020 IFZB=1ORZ0<>0ORB$<>" "THEN2222
2025 IFK=9Z0=5
2030 IFK=1Z0=1
2040 IFK=2Z0=2
2050 IFK=3Z0=3
2060 IFK=4Z0=4
2070 IFK=5Z0=6
2060 IFK=6ZQ=7
2065 IFZQ=7ANDZ8=1THEN2222
   3 GOTO2000
     GOSUB3000:C2=C2+1:A2$(C2,1)=M1$:A2$(C2,2)=M2$:A2$(C2,3)=M3$:A2$(C2,4)=M4$:A
2$(C2,5)=M6$:A2$(C2,6)=M7$:A2$(C2,7)=M5$:A2$(C2,8)=Z0$:A9(C2)=ZZ:ZZ=0:Z0$=" "
2155 M1$=" ":M2$=" ":M3$=" ":M4$=" ":M5$=" ":M6$=" ":M7$=" ":RETURN
2222 PRINT"I DO NOT UNDERSTAND THIS SENTENCE!":GOTO1140
3000 FORI6=1TOC1:IFM7$=A0$(I6)THENI7=A1(I6,3):M7$=A0$(I7)
3010 NEXTI6:RETURN
3070 IFZI=9M5$=M5$+"/"+D$
3075 IFZI=5M6$=M6$+"/"+D$
3080 IFZI=2M2$=M2$+"/"+D$
3090 IFZI=3M3$=M3$+"/"+D$
3100 IFZI=4M4$=M4$+"/"+D$
3110 D$=" ":RETURN
3180 IFM7$=" "THENM7$=C$ELSEM7$=M7$+" "+C$
3190 ZH=N:ZI=K:RETURN
4100 IFN=1ORN=5THEN1344
4110 IFZH=9ANDN=4ZH=3:GOTO1344
4120 GOTO2222
4300 ZA=ZA+1:IFK=1M1$=M1$+"/"+C$
4320 IFK=2M2$=M2$+"/"+C$
4330 IFK=4M4$=M4$+"/"+C$
   3 RETURN
   ) S1$=S3$:S2$=S4$:V1=V3:V2=V4:X1=X3:X2=X4:S3$=" ":S4$=" ":V3=0:V4=0:X3=0:X4=0
:RETURN
4700 S3$=E$:V3=T:X3=XM:GOSUB4900:RETURN
4800 S4$=E$:V4=T:X4=XM:GOSUB4900:RETURN
4900 IFS1$<>" "F$=S1$:F1=V1:F2=X1:GOSUB5000
4910 IFS2$<>" "F$=S2$:F1=V2:F2=X2:GOSUB5000
4920 RETURN
5000 IFM6$<>" "ORM3$<>" "ORM5$<>" "RETURN
5001 IFF2=1ANDXM=1RETURN
5002 IFZD>0RETURN
5005 C3=C3+1:A3$(C3,1)=F$:A3$(C3,2)=E$:A4(C3,1)=F2:A4(C3,2)=XM
5010 A8!(C3,1)=F1:A8!(C3,2)=T:RETURN
6000 IFO$<>" "M6$=M6$+"/"+O$
6001 IFZH=9THENGOSUB3070
6002 IFZQ>0THEN6200
6005 IFM7$=" "THEN2222ELSEGOSUB2135:GOTO1140
6200 IFZQ=7THEN7000ELSEL=1:P=0
6205 IFC1<1THEN6210
6206 FORI8=1TOC1:IFLEN(M7$)<>LEN(A0$(I8))ORA0$(I8)<>M7$THEN6208
6207 B2=A1(I8,3):IFB2>0THENM7$=A0$(B2)
   9 NEXTI8
```

```
6210 L=1:P=0
6220 IFC2<1THEN6700
6230 FORIB=1TOC2
6300 IFM1$=" "THEN6320
6310 MA$=A2$(IB,1):MB$=M1$:GOSUB9700:IFMG=0THEN6600
6320 IFM2$=" "THEN6340
6330 MA$=A2$(IB,2):MB$=M2$:GOSUB9700:IFMG=0THEN6600
6340 IFM3$=" "THEN6360
6350 MA$=A$(L+3):MB$=M3$:GOSUB9700:IFMG=0THEN6600
6360 IFM4$=" "THEN6380
6370 MA$=A2$(IB,4):MB$=M4$:GOSUB9700:IFMG=0THEN6600
6380 IFM5$=" "THEN6400
6390 MA$=A2$(IB,7):MB$=M5$:GOSUB9700:IFMG=0THEN6600
6400 IFM6$=" "THEN6420
6410 MA$=A2$(IB,5):MB$=M6$:GOSUB9700:IFMG=0THEN6600
6420 IFM7$=A2$(IB,6)THEN6460ELSE6600
6450 MA$=A2$(IB,3):MB$=M3$:GOSUB9700:IFMG=0THEN6600
6460 P=IB
6600 NEXTIB
6700 IFZQ<8ANDP=0THEN7920
6705 IFZQ=8ANDP=0PRINT"NO":GOTO1140
6720 IFZQ=A9(P)THENPRINTA2$(P,8):GOTO1140
6730 IFZQ=1ANDA2$(P,1)<>" "THENPRINTA2$(P,1):GOTO1140
6740 IFZQ=2ANDA2$(P,2)<>" "THENPRINTA2$(P,2):GOTO1140
6750 IFZQ=3ANDA2$(P,3)<>" "THENPRINTA2$(P,3):GOTO1140
6760 IFZQ=4ANDA2$(P,4)<>" "THENPRINTA2$(P,4):GOTO1140
6770 IFZQ=5ANDA2$(P,7)<>" "THENPRINTA2$(P,7):GOTO1140
6780 IFZQ=6ANDA2$(P,5)<>" "THENPRINTA2$(P,5):GOTO1140
6800 IFZQ=8PRINT"YES":GOTO1140
6810 GOTO7920
7000 IFX1=1ORV1>0ORX3=1ORZY>2ORM3$<>" "ORM5$<>" "ORM6$<>" "ORS1$=" "ORS3$=" "THE
N2222
7001 IK=10:IL=4
7002 L=1:FORN=1TOIK:FORK=1TOIL:H(N,K)=0:NEXTK:NEXTN:KI=0:N=0:ZD=0
7003 IFC3<1THEN7920
7004 L=0
7015 GOSUB7900
7020 IFSR=9THEN7060
7022 P=0:J=0
7025 IFA3$(L,1)=S3$THENP=2:J=1
7026 IFA3$(L,2)=S3$THENP=1:J=2
7030 IFP=0ORJ=0THEN7015
7031 IFA4(L,P)=1ORA4(L,J)=1THEN7015
7032 IFA8!(L,J)>1THEN7015
7035 IFKI=0THEN7043
```

```
7037 T=0
7039 T=T+1:ZZ=H(T,1):ZQ=H(T,4):IFA3$(ZQ,ZZ)=A3$(L,P)THEN7044
7041 IFT<KITHEN7039
7043 KI=KI+1:T=KI
7044 IFKI>IKTHEN7925
7045 ZA=A8!(L,P):IFZA>0THENZD=T
7047 H(T,1)=P:H(T,2)=ZA:H(T,4)=L:IFA3$(L,P)=S1$THENN=T
7049 GOTO7015
7060 IFN=0ORKI=0THEN7920
7062 ZI=N:ZA=H(N,2):IFZA>0THEN7930
7100 IFZD=0THEN7920
7110 FORP=1TOKI:T=H(P,1):ZQ=H(P,4):IFH(P,2)>0THEN7115ELSE7260
7115 ZE=0:FORN=1TOKI:K=H(N,1):K9=H(N,4):IFN=PTHEN7250ELSEH(N,3)=0
7130 L=0
7140 GOSUB7900:ZZ=0:ZB=0:IFSR=9THEN7240
7160 IFA3$(ZQ,T)=A3$(L,1)THENZZ=1:ZB=2
7170 IFA3$(ZQ,T)=A3$(L,2)THENZZ=2:ZB=1
7180 IFZZ=0THEN7140
7190 IFA3$(L,ZB)=A3$(K9,K)THEN7200ELSE7140
7200 ZA=A8!(L,ZZ):ZB=A8!(L,ZB):IFZA>0ANDZB=0THEN7230ELSE7140
7230 H(N,3)=ZA:GOTO7140
7240 IFH(N,3)>0ZE=ZE+1
7250 NEXTN:IFZE=KI-1THEN7270
7260 NEXTP:GOTO7920
7270 FORN=1TOKI:IFP=NTHEN7290
7280 IFH(N,2)>0THENH(N,2)=H(N,2)*H(N,3)ELSEH(N,2)=H(N,3)
7290 H(N,3)=0:NEXTN
7300 FORN=1TOKI:IFN=PTHEN7420ELSET=H(N,1)
7305 ZQ=H(N,4)
7310 FORK=1TOKI:IFK=PORK=NTHEN7410
7320 L=1:ZZ=H(K,1)
7322 L=0
7325 K9=H(K,4)
7330 GOSUB7900:ZC=0:ZD=0
7340 IFSR=9THEN7410
7350 IFA3$(ZQ,T)=A3$(L,1)THENZC=1:ZD=2
7360 IFA3$(ZQ,T)=A3$(L,2)THENZC=2:ZD=1
7370 IFZC=0THEN7330
7380 IFA3$(K9,ZZ)=A3$(L,ZD)THEN7390ELSE7330
7390 IFA4(L,1)=1ORA4(L,2)=1THEN7400ELSE7330
7400 H(N,3)=A8!(L,ZC):H(K,3)=A8!(L,ZD):GOTO7330
7410 NEXTK
7420 NEXTN
7430 FORN=1TOKI:IFN=PTHEN7450
7440 IFH(N,3)>0H(N,2)=H(N,2)*H(N,3)
7450 NEXTN
7460 T=0:FORN=1TOKI:IFN<>PTHENT=T+H(N,2):NEXTN
7470 ZA=H(P,2)/T:IFH(ZI,3)>0THENZA=ZA*H(ZI,3)
7480 GOTO7930
7500 IFC1<1THENRETURN
7502 MB$=" ":P=0:B2=0
7505 INPUT"VERB PHRASE";MB$
7510 FORI6=1TOC1:IFA0$(I6)=MB$THEN7520
7515 NEXTI6:PRINT"UNKNOWN!":RETURN
7520 I7=A1(I6,4):IFI7>0THEN7540
7521 IFA1(I6,1)<>2THENPRINT"NOT A VERB!":RETURN
7522 PRINT"NEW ACTION VERB PROC"
7525 B9=C9+1:IFB9+2>19THENPRINT"OUT OF MEMORY":END
7530 C9=B9:I7=B9:A1(I6,4)=B9
7532 B4=0:INPUT"# INSTRUCTION SPACES";B4:IFB4<2THEN7432
7533 IFC9+B4-1>19THENPRINT"INSUFFICIENT MEMORY":GOTO7532
7534 A7(I7)=B4-1:B6=I7:FORB5=1TOB4-1:B6=B6+1:A7(B6)=0:NEXTB5:PRINT"INITIALIZED T
0 NOPS"
7535 C9=C9+B4-1
7540 INPUT"L=LIST,C=CHANGE,S=STRING,E=END";MA$
7545 IFMA$="E"THENRETURN
7546 IFMA$="S"THEN7575
7550 INPUT"FROM,TO LOCATIONS";B1,B2:IFB1>B2ORB1<1ORB2>A7(I7)THEN7550
7553 IFMA$="C"THEN7565
7555 FORB3=B1TOB2:PRINT" LOC ";B3;" CONTENTS ";A7(I7+B3);
7556 IFA7(I7+B3)>=0ANDA7(I7+B3)<=255THENPRINT" ASC ";CHR$(A7(I7+B3))ELSEPRINT" "
7557 NEXTB3:GOTO7540
7565 FORB3=B1TOB2:PRINT" LOC ";B3;" WAS ";A7(I7+B3);
7566 IFA7(I7+B3)>=0ANDA7(I7+B3)<=255THENPRINT" ASC ";CHR$(A7(I7+B3))ELSEPRINT" "
7567  INPUT"NOW";A7(I7+B3)
7570 NEXTB3:GOTO7540
7575 INPUT"LENGTH POINTER LOCATION";B1:IFB1<1ORB1>A7(I7)THEN7575
7580 MA$=" ":INPUT"STRING VALUE";MA$:B2=LEN(MA$):IFB2<1ORB2+B1>A7(I7)THEN7580
7582 A7(I7+B1)=B2
7585 FORB3=1TOB2:A7(I7+B1+B3)=ASC(MID$(MA$,B3,1)):NEXTB3:GOTO7540
7900 L=L+1:IFL>C3THENSR=9:L=0ELSESR=0
7915 RETURN
7920 PRINT"UNKNOWN":GOTO1140
7925 PRINT"TOO COMPLEX":GOTO1140
7930 IFV3>0ZA=ZA*V3
7940 PRINTZA:GOTO1140
8000 IFC1>0ORC2>0ORC3>0ORC9>0THEN8005ELSEEND
8005 INPUT"STORE DATA BASE ON TAPE(Y/N)";V$
8010 IFV$<>"Y"THENEND
8021 BN=INT(C9/I7)+1
```

```
8022 PRINT#-1,C1,C2,C3,BH
8025 IFC1<1THEN8035
     FORI6=1TOC1
     PRINT#-1,A0$(I6),A1(I6,1),A1(I6,2),A1(I6,3),A1(I6,4)
8033 NEXTI6
8035 IFC2<1THEN8045
8040 FORI6=1TOC2
8042 PRINT#-1,A2$(I6,1),A2$(I6,2),A2$(I6,3),A2$(I6,4),A2$(I6,5),A2$(I6,6),A2$(I6
,7),A2$(I6,8),A9(I6)
8043 NEXTI6
8045 IFC3<1THEN8055
8050 FORI6=1TOC3
8052 PRINT#-1,A3$(I6,1),A3$(I6,2),A4(I6,1),A4(I6,2),A8!(I6,1),A8!(I6,2)
8053 NEXTI6
8055 IFBH<1THENEND
8060 FORI6=1TOBH
8062 I7=0
8065 B0=-1:I7=I7+1:IFI7<=C9THENB0=A7(I7)
8070 B1=-1:I7=I7+1:IFI7<=C9THENB1=A7(I7)
8075 B2=-1:I7=I7+1:IFI7<=C9THENB2=A7(I7)
8080 B3=-1:I7=I7+1:IFI7<=C9THENB3=A7(I7)
8085 B4=-1:I7=I7+1:IFI7<=C9THENB4=A7(I7)
8090 B5=-1:I7=I7+1:IFI7<=C9THENB5=A7(I7)
 195 B6=-1:I7=I7+1:IFI7<=C9THENB6=A7(I7)
     B7=-1:I7=I7+1:IFI7<=C9THENB7=A7(I7)
8105 B8=-1:I7=I7+1:IFI7<=C9THENB8=A7(I7)
8110 B9=-1:I7=I7+1:IFI7<=C9THENB9=A7(I7)
8115 BA=-1:I7=I7+1:IFI7<=C9THENBA=A7(I7)
8120 BB=-1:I7=I7+1:IFI7<=C9THENBB=A7(I7)
8125 BC=-1:I7=I7+1:IFI7<=C9THENBC=A7(I7)
8130 BD=-1:I7=I7+1:IFI7<=C9THENBD=A7(I7)
8135 BE=-1:I7=I7+1:IFI7<=C9THENBE=A7(I7)
8140 BF=-1:I7=I7+1:IFI7<=C9THENBF=A7(I7)
8145 BG=-1:I7=I7+1:IFI7<=C9THENBG=A7(I7)
8150 PRINT#-1,B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,BA,BB,BC,BD,BE,BF,BG
8155 NEXTI6:END
8500 IFC2<1THENRETURN
8502 FORI6=1TOC2:CLS:PRINT"WHEN-";A2$(I6,3):PRINT"WHY-";A2$(I6,7)
8505 PRINT"WHO-";A2$(I6,1):PRINT"WHAT-";A2$(I6,2):PRINT"WHERE-";A2$(I6,4)
8510 PRINT"VERB-";A2$(I6,6):PRINT"HOW-";A2$(I6,5):PRINT"OBJ CLAUSE-";A2$(I6,8)
8515 INPUT"HIT ENTER TO CONTINUE";V$
8520 NEXTI6:RETURN
9700 MG=0:IFLEN(MA$)<LEN(MB$)RETURN
9705 IFLEFT$(MB$,2)=" /"MB$=MID$(MB$,3,LEN(MB$)-2)
9710 FORMF=1TOLEN(MA$)-LEN(MB$)+1
```

```
9720 IFMB$=MID$(MA$,MF,LEN(MB$))MG=1:RETURN
9730 NEXTMF:RETURN
9800 FORL=0TO14:A6$(L)=" ":A5!(L)=0:NEXTL:A6$(0)=V$:GT=0:LT=0:EQ=0:RT=0
9801 I7=A1(59,4):P=I7
9802 I6=A7(I7)
9803 I7=I7+1:L=A7(I7)
9804 IFL=0THEN9803
9806 IFL=1THENCLS:GOTO9803
9807 IFL=2THENGOSUB9970:V$="":FORK=1TOSR:I7=I7+1:N=A7(I7):V$=V$+CHR$(N):NEXTK:A6
$(L)=V$:GOTO9803
9808 IFL=3THENGOSUB9970:V$="":FORK=1TOSR:I7=I7+1:N=A7(I7):V$=V$+CHR$(N):NEXTK:A5
!(L)=VAL(V$):GOTO9803
9809 IFL<>4THEN9815
9810 GOSUB9970:GT=0:LT=0:EQ=0
9811 IFA6$(L)=A6$(SR)THENEQ=1
9812 IFA6$(L)>A6$(SR)THENGT=1
9813 IFA6$(L)<A6$(SR)THENLT=1
9814 GOTO9803
9815 IFL<>5THEN9821
9816 GOSUB9970:GT=0:LT=0:EQ=0
9817 IFA5!(L)=A5!(SR)THENEQ=1
9818 IFA5!(L)>A5!(SR)THENGT=1
9819 IFA5!(L)<A5!(SR)THENLT=1
9820 GOTO9803
9821 IFL=6THENGOSUB9970:N=A5!(L):K=A5!(SR):POKEN,K:GOTO9803
9822 IFL=7THENGOSUB9970:A5!(L)=PEEK(A5!(SR)):GOTO9803
9823 IFL=8THENGOSUB9970:A5!(L)=A5!(L)+A5!(SR):GOTO9803
9824 IFL=9THENGOSUB9970:A5!(L)=A5!(L)-A5!(SR):GOTO9803
9825 IFL=10THENGOSUB9970:A5!(L)=A5!(L)*A5!(SR):GOTO9803
9826 IFL=11THENGOSUB9970:A5!(L)=A5!(L)/A5!(SR):GOTO9803
9827 IFL=12THENGOSUB9970:A5!(L)=A5!(L)[A5!(SR):GOTO9803
9828 IFL=13THENGOSUB9970:A6$(L)=A6$(L)+A6$(SR):GOTO9803
9829 IFL=14THENGOSUB9970:N=INT(A5!(SR)):A5!(L)=N:GOTO9803
9830 IFL=15THENGOSUB9970:A5!(L)=SIN(A5!(SR)):GOTO9803
9831 IFL=16THENGOSUB9970:A5!(L)=COS(A5!(SR)):GOTO9803
9832 IFL=17THENGOSUB9970:A5!(L)=TAN(A5!(SR)):GOTO9803
9833 IFL=18THENGOSUB9970:A5!(L)=ATN(A5!(SR)):GOTO9803
9834 IFL=19THENGOSUB9970:A5!(L)=VAL(A6$(SR)):GOTO9803
9835 IFL=20THENGOSUB9970:A6$(L)=STR$(A5!(SR)):GOTO9803
9836 IFL=21THENGOSUB9968:A6$(K)=STRING$(A5!(L),A6$(SR)):GOTO9803
9837 IFL=22THENGOSUB9972:LPRINTA6$(SR):GOTO9803
9838 IFL=23THENGOSUB9972:A6$(SR)=INKEY$:GOTO9803
9839 IFL<>24THEN9845
9840 GOSUB9972:I7=P:FORN=1TOI6
9841 I7=I7+1:IFA7(I7)<>999THEN9844
```

```
9842 I7=I7+1:IFA7(I7)<>SRTHEN9844
9843 GOTO9803
9844 NEXTN:PRINT"INVALID TAG ";SR:GOTO1140
9845 IFL=25THENGOSUB9970:A5!(L)=ABS(A5!(SR)):GOTO9803
9846 IFL=26THENGOSUB9970:A5!(L)=INP(A5!(SR)):GOTO9803
9847 IFL=27THENGOSUB9972:PRINTA6$(SR):GOTO9803
9848 IFL=28THENGOSUB9970:PRINT@A5!(L),A6$(SR):GOTO9803
9849 IFL=29THENGOSUB9972:INPUTA6$(SR):GOTO9803
9850 IFL=30THENGOSUB9972:INPUT#-1,A6$(SR):GOTO9803
9851 IFL=31THENGOSUB9972:PRINT#-1,CHR$(34)+A6$(SR)+CHR$(34):GOTO9803
9852 IFL=32THENGOSUB9968:A6$(K)=LEFT$(A6$(L),A5!(SR)):GOTO9803
9853 IFL=33THENGOSUB9968:A6$(K)=RIGHT$(A6$(L),A5!(SR)):GOTO9803
9854 IFL=34THENGOSUB9966:A6$(N)=MID$(A6$(K),A5!(L),A5!(SR)):GOTO9803
9855 IFL=35THENGOSUB9970:A5!(L)=A5!(SR):GOTO9803
9856 IFL=36THENGOSUB9970:A6$(L)=A6$(SR):GOTO9803
9857 IFL=37THENGOSUB9970:A5!(L)=EXP(A5!(SR)):GOTO9803
9858 IFL=38THENGOSUB9970:A5!(L)=LOG(A5!(SR)):GOTO9803
9860 IFL<>39THEN9863
9861 IFEQ=1THEN9840
9862 I7=I7+1:GOTO9803
9863 IFL<>40THEN9865
9864 IFGT=1THEN9840ELSE9862
9865 IFL<>41THEN9867
9866 IFLT=1THEN9840ELSE9862
9867 IFL=42THENGOSUB9970:A5!(L)=ASC(A6$(SR)):GOTO9803
9868 IFL=43THENGOSUB9970:A6$(L)=CHR$(A5!(SR)):GOTO9803
```

```
9869 IFL=44THENGOSUB9970:A5!(L)=LEN(A6$(SR)):GOTO9803
9870 IFL=45THENGOSUB9970:A5!(L)=SQR(A5!(SR)):GOTO9803
9871 IFL=46THENI7=RT:RT=0:GOTO9803
9872 IFL=47THENGOSUB9970:PRINT@A5!(L),A5!(SR):GOTO9803
9873 IFL=48THENGOSUB9970:N=A5!(L):K=A5!(SR):SET(N,K):GOTO9803
9874 IFL=49THENGOSUB9970:N=A5!(L):K=A5!(SR):RESET(N,K):GOTO9803
9876 IFL=51THENGOSUB9970:N=A5!(L):K=A5!(SR):OUTN,K:GOTO9803
9877 IFL=52THENGOSUB9972:A5!(SR)=RND(0):GOTO9803
9878 IFL<>53THEN9885
9880 GOSUB9968:N=A5!(SR):MA$=A6$(K):MB$=A6$(L)
9882 GOSUB9700:A5!(N)=MG
9884 GOTO9803
9885 IFL=54THEN1140
9886 IFL=55THENRT=I7+1:GOTO9840
9887 IFL<>50THEN9890
9888 GOSUB9968:N=A5!(L):SL=A5!(SR):IFPOINT(N,SL)THENA5!(K)=1ELSEA5!(K)=0
9889 GOTO9803
9890 IFL=999THENI7=I7+1:GOTO9803
9899 PRINT"INVALID OP":N=I7-P:PRINT"LOC ";N;" CODE ";L
9900 GOTO1140
9966 I7=I7+1:N=A7(I7)
9968 I7=I7+1:K=A7(I7)
9970 I7=I7+1:L=A7(I7)
9972 I7=I7+1:SR=A7(I7)
9974 RETURN
```

JOIN THE 1000'S WHO ARE ENJOYING

THE 80 NOTEBOOK

THE MONTHLY JOURNAL FOR ALL TRS-80 USERS

WITH THESE EXCITING FEATURES AND DEPARTMENTS: (AND MORE!!)

* SCIENTIFIC SOFTWARE
* LETTERS TO THE EDITOR
* PRACTICAL APPLICATIONS
* ENTERTAINMENT PROGRAMS
* WORD PROCESSING SYSTEMS
* ARTIFICIAL INTELLIGENCE
* SYSTEM UTILITY SOFTWARE
* RADIO SHACK NEWS RELEASES
* DATA BASE MANAGEMENT SYSTEMS
* EDUCATIONAL AND CAI PROGRAMMING
* SIMULATIONS AND COMPUTER MODELING
* GRAPHICS AND ANIMATION TECHNIQUES
* ASSEMBLY LANGUAGE PROGRAMMING LESSONS
* LEVEL I, II AND DISK BASIC PROGRAMMING LESSONS
* OPERATING SYSTEMS, LANGUAGES AND COMPILER DESIGN
* PROGRAM LISTINGS (UP TO 24 PAGES IN EVERY ISSUE!!)
* ARTICLES DEALING WITH UNUSUAL AND INTERESTING USES FOR YOUR TRS-80

* GAMES
* PUZZLES
* CONTESTS
* GAMBLING
* NEW PRODUCTS
* TRS-80 CLUB NEWS
* PERSONAL FINANCE
* SOFTWARE EXCHANGE
* BUSINESS SOFTWARE
* SORTING TECHNIQUES
* FREE CLASSIFIED ADS
* PRODUCT EVALUATIONS
* ARTICLES ON HARDWARE

NOTE: The term "TRS-80" is a registered trademark of Radio Shack, a Division of Tandy Corporation, who is not affiliated with THE 80 NOTEBOOK in any way.

THE 80 NOTEBOOK
R. D. #3
Nazareth, PA 18064
Phone: 215-759-6873

NAME ______________________
ADDRESS ____________________
____

CHECK ONE: NEW ______ RENEWAL ______
1 YEAR SUBSCRIPTION: $14 __________
2 YEAR SUBSCRIPTION: $26 __________
CANADA/MEXICO: ADD $5/ YEAR

3 YEAR SUBSCRIPTION: $36 __________
SAMPLE COPY: $2 _____ AIR MAIL $4 _____
FOREIGN: ADD $12/ YEAR

AVOID THE 1980 PRICE INCREASES - SUBSCRIBE TODAY!

THE 80 NOTEBOOK
R.D.#3
Nazareth, PA 18064

4/80

3/81